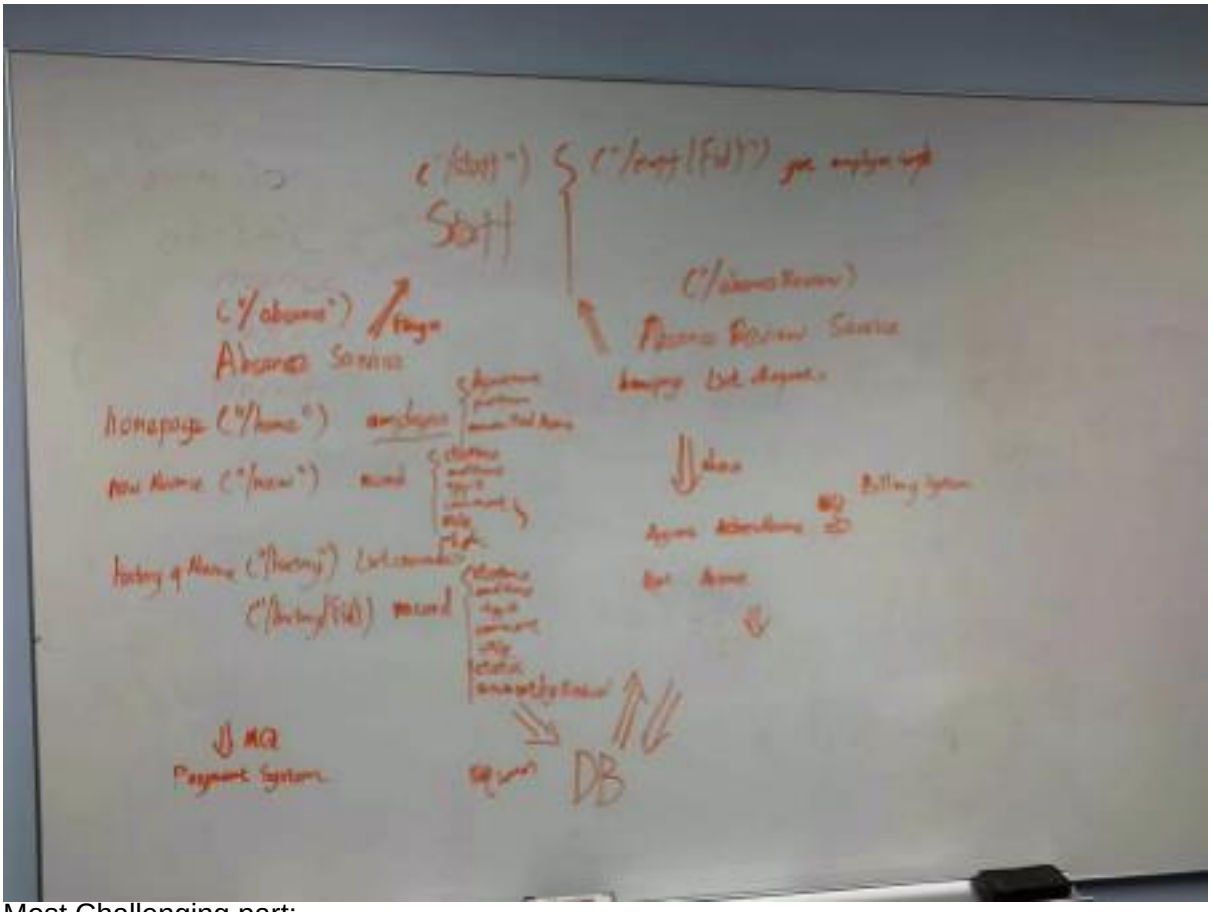
Project 1:

FINEOS specializes in employee benefits. It provides a multifunctional platform, which we called AdminSuite, for the insurers to deal with everything related to the employee's benefits. For example, it consists of subcomponents like the Claims system, to provide accurate, swift claims management, and like the Billing system, which empowers insurers to quickly adapt their solution to support new insurance products. My project basically focuses on another component, FINEOS Absence, which gives employers an integrated holistic solution for absence management. This project provides an interface to reduce manual calculations across systems and spreadsheets, it is easy to use and friendly to both employees and employers.

Tech stack
For the backend part
Backend: ASP.NET Core, Entity Framework
Database: Azure SQL server (MS SQL Server)
CI/CD: Azure DevOps
Frontend: Angular 6, HttpClient to communicate with the backend.

The project consists of two parts, one for the employee users and one for the employer users. For the employee side, they will be able to view their overall attendance situation like how many paid absence days they have and the history of past records. Employees will need to submit a record for the week when they have an absence. Employers can review all the attendance records submitted by employees. They can reject the submitted record and leave some comments to let employees know if something went wrong, and the employee will receive an email notification about the comment and re-submit the form. They can also approve the attendance record and forward it to the other systems like the claim system and billing system for other operations.



Most Challenging part:

The most challenging part of the project is to improve the fault tolerance. Such as dealing with cascading failures to prevent it turning our application down by adding circuit breaker between services, it will track success and fail calls over time. When failure reaches some threshold, any call to the further service will immediately get an error. It will periodically allow one call to the further service to check availability.

Project 2:

charterUP offers private shuttle solutions for organizations of any size and provides different solutions. charterUP contributes a lot in encouraging individuals and organizations to make low-carbon travel through commuting by shuttle. My team focused on building new features specific to the users from universities and campus. The users will need to provide detailed information such as routes, the type of shuttle they want, the frequency of the shuttle. Then they can keep track of and manage the shuttle plan they have.

The project was based on the ASP.NET. We have three core services which are route service, time service and shuttle service, which route service used to customize the route and stop points, time service used to customize the start time, end time and frequency of each shuttle

line, shuttle service responsible for the shuttle configuration.

For the backend part
Backend: ASP.NET Core, ADO.NET
Database: Azure SQL server (MS SQL Server)
CI/CD: Azure DevOps
Frontend: Angular 6, HttpClient to communicate with the backend.